Title

RED-BL: Evaluating dynamic workload relocation for data center networks

Authors

Ilyas, Muhammad Saqib
Raza, Saqib
Chen, Chao-Chih
et al.

Publication Date

2014-10-01

DOI

10.1016/j.comnet.2014.07.001

RED-BL: Evaluating Dynamic Workload Relocation for Data Center Networks

Muhammad Saqib Ilyas[a,*], Saqib Raza[b], Chao-Chih Chen[c], Zartash Afzal Uzmi[a], Chen-Nee Chuah[c]

[a] *SBA School of Science and Engineering, LUMS, Lahore, Pakistan*
[b] *Cisco Systems, Inc., USA*
[c] *University of California, Davis, CA, USA*

Abstract

In this paper, we present RED-BL (Relocate Energy Demand to Better Locations), a framework to minimize the electricity cost for operating data center networks over consecutive intervals of fixed duration. Within each interval, RED-BL provides a mapping of workload to a set of geographically distributed data centers. To this end, RED-BL uses the geographical and temporal variations in electricity prices as exhibited by electrical energy markets. In addition, we incorporate the transition costs associated with a change in workload mapping from one interval to the next, over a planning window comprising multiple such intervals. This results in a sequence of workload mappings that is optimal over the entire planning window, even though the workload mapping in a given interval may not be locally optimal.

Our evaluation of RED-BL uses electricity prices from the US markets and workload traces from live Internet applications with millions of users. We find that RED-BL can reduce the electric bill by as much as 45% compared to the case when the workload is uniformly distributed. When compared to existing workload relocation solutions, for a wide range of data center deployment sizes, RED-BL achieves electricity cost savings that are 8.28% higher, on average. This seemingly modest reduction can save millions of dollars for the operators. The cost of this saving is an inexpensive computation at the start of each planning window.

Keywords: Data center, Electricity cost, Optimization, Workload relocation, Configuration planning

*Corresponding author

Email addresses: saqibm@lums.edu.pk (Muhammad Saqib Ilyas), sraza@ucdavis.edu (Saqib Raza), cchchen@ucdavis.edu (Chao-Chih Chen), zartash@lums.edu.pk (Zartash Afzal Uzmi), chuah@ucdavis.edu (Chen-Nee Chuah)

1. Introduction

Geo-diverse data centers enable robust and low-latency cloud services. The electricity cost for this huge infrastructure is a significant fraction of the operational cost (15%) [1] as well as capital cost [2]. Due to increasing demand for cloud services and increasing electricity prices, it is essential for data center operators to cut their electricity bill [3, 4].

A data center's electricity bill for a particular period of time is the product of the unit price of electricity and the amount of electrical energy consumed. Hence, two possibilities to reduce data center electricity costs are: i) use cheaper sources of electricity, and ii) reduce energy consumption. Our present work jointly exploits both of these dimensions.

A geo-diverse data center deployment is dimensioned according to peak workload, which occurs for only a short period of time [5]. Therefore, most of the time, the workload may be mapped to a subset of the over-provisioned infrastructure. During a given period of time, a data center's energy consumption is an affine function of the workload it handles [6]. Therefore, geographic diversity in electricity prices [2] may be leveraged to cut electricity cost by directing most of the workload to data centers with cheaper electricity prices.

In addition to geographic diversity, electricity prices exhibit temporal diversity as well [2] causing the cheapest set of data centers to handle current workload to change with time. Therefore, an electricity cost reduction technique for geo-diverse data centers must periodically update its choice of data centers to be used for handling the current workload. We define an *interval* as a period of time for which electricity prices are fixed and workload is known. We also define a sequence of consecutive intervals as a *planning window*. An interval may be an hour long and the planning window may be 24 hours, for instance. The problem, then, is to pick the distribution, or *mapping*, of workload to data centers for each interval in a planning window to minimize the electricity cost by exploiting the geo-temporal variation in electricity prices and temporal variations in workload. If we define the aggregation of workload mapping for all data centers during a particular interval as a *network state*, this problem may be viewed as determining a state trajectory that is electricity cost optimal over the corresponding planning window.

The electricity cost savings resulting from workload relocation are somewhat limited due to lack of energy proportionality in today's data centers [5]. Therefore, it was proposed to dynamically scale the active infrastructure in response to changes in the magnitude of workload [5, 7]. This capacity scaling is expected to incur some overhead electricity cost which may be modeled as the *state transition cost* in the state trajectory problem.

Theoretically, one can benefit from capacity scaling by shutting down elastic load when it is not needed. This scheme minimizes the electricity consumption, but conventional wisdom suggests that restarts affect equipment lifetime and operators are generally reluctant to adopt this approach. On the other extreme, elastic load may be left in an idle state, but the reduction in electricity consumption would be quite small. In between these two extremes would be

Dynamic Voltage and Frequency Scaling (DVFS) techniques. The transition cost would be zero for the idling scheme since no state-change overhead is incurred. The transition costs would be quite high if elastic load is shutdown (and restarted later when needed), whereas DVFS would account for transition costs somewhere between the two extremes [8]. In this work, we experiment with the entire spectrum of transition costs in order to generalize our results.

To the best of our knowledge, prior work has largely taken a micro-scale view of this problem by scaling the data center capacity at the granularity of states of individual servers within a data center [9, 10, 11, 12, 13, 7] and, in some cases, has altogether ignored transition costs resulting from capacity scaling. These approaches lack scalability to multi-data center scenarios or are sub-optimal. In this paper, we address the challenges of scalability as well as incorporation of transition costs into the optimization problem.

In the present work, we approach a scalable solution to this problem by treating all the elastic electric load[1] in a data center as an aggregate and determining this aggregate's state for each interval in a planning window. For every interval in the planning window, our coarse-granularity formulation provides the average utilization of the servers within a data center. Relaxing a discrete optimization problem to a continuous one typically introduces an approximation error. The magnitude of this error is expected to be small for large scale problems such as geo-diverse data centers. Determination of the approximation error's magnitude is beyond the scope of this paper and is left as future work.

To motivate the significance of transition costs in the dynamic scaling of geo-diverse data center capacity and our scheme for incorporating the transition costs to the optimization problem, we will use a simple example shown in Figure 1. It depicts an example instance of the state trajectory problem for a planning window consisting of three intervals represented along the horizontal axis. For each interval, we show three sample states represented using rounded rectangles. Each state is labeled with a name in the lower left corner and the corresponding electricity cost in the lower right corner. Moreover, transition between states in consecutive intervals is shown using arrows and the corresponding transition cost is shown as a label over the arrow. We consider three data centers in this example, represented using circles numbered 1, 2 and 3. In Figure 1, the relative height of the circles in a given interval represents the diversity in electricity prices. For instance, in interval 3, data center 3 has the lowest electricity price. In a particular state, the workload mapped to each data center is represented using shading within the circle. For simplicity of demonstration, we assume that the cumulative workload is fixed at a value that equals 1.3 times a single data center's workload capacity.

In the absence of transition costs, the optimal state trajectory could be obtained by making a *greedy* choice of state in each interval (the path S2→S6→S8 in Figure 1) [2, 7, 13]. This is clearly the lowest possible sum of state costs

[1]The electric load that may be turned on or off to save electricity cost without long delays or long-term impact on performance.

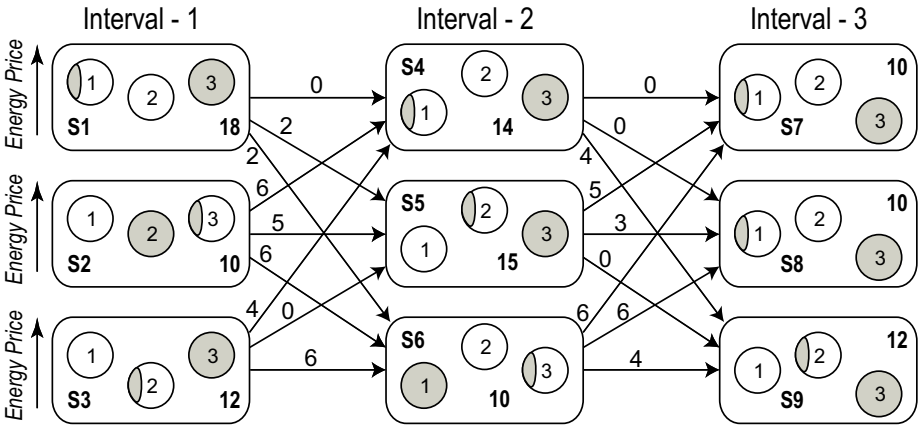

Figure 1: A motivating example that depicts the workload-mapping problem for three consecutive intervals involving three data centers of equal capacity. For this example, the workload is assumed to be constant equal to 1.3 times the capacity of a single data center, in all three intervals. Of many possible states in each interval, we show just three example candidate states along with the electricity cost for being in those states. Cost of transition from one state to another in the next interval are also labeled on the arrows representing the state transition.

without considering any transition costs. With transition costs included, however, the greedy solution yields a total cost of 42. We refer to such a strategy as Relocate Energy Demand to Cheaper Locations (RED-CL).

One may also consider a *static* deployment configuration where an operator selects the data centers that have the lowest average electricity price over the planning window. This corresponds to the path S1→S4→S7, with the sum of state costs equal to 42. Since the workload mapping does not change, there are no transition costs, and hence the total solution costs is also 42. In general, depending on the magnitude of transition costs, the static solution could be better or worse compared to the greedy solution.

The optimal solution from Fig. 1 is the path S3→S5→S9, with a total cost of 39. For this state path, the sum of state costs is 39, which is higher than the corresponding component for the greedy solution. However, the sum of transition costs is 0 resulting in an overall lower total solution cost than both the static and the greedy strategy. This simple example illustrates that it is important to consider the costs associated with relocating demands in operational data centers.

Our present work uses the overhead incurred in changing the state of the elastic load within a data center as the transition costs. However, in practice, there can be other forms of transition costs as well, which would vary from one deployment to the other. Examples of other sources of transition costs include, but may not be limited to, the following:

- **Convergence time:** An operator might change the way user traffic is routed to data centers at various layers of the network stack. For in-

stance, this could be done by modifying DNS entries or BGP routing tables. However, these and other workload rerouting schemes would have non-negligible convergence times. For instance, many DNS caches do not honor the time-to-live (TTL) values for DNS entries [14]. Also, BGP routing table changes have been shown to take an unpredictable amount of time to reflect globally [15]. Thus, if data center A is scaled down for interval $j+1$ and it's workload is redirected to data center B, the former may continue to receive some fraction of the workload during interval $j+1$. Some *reserve* capacity must, therefore, be kept active at data center A during interval $j+1$, thereby reducing the electricity cost savings.

- **Consistency traffic:** In order for any request to be handled anywhere, the data store for the applications must be replicated and consistency must be maintained. The cost of inter-data center traffic is quite high, hence this form of transition costs may be quite significant. The magnitude of such overheads is not easy to predict because replication schemes are operator and application dependent. To the best of our knowledge, the current body of knowledge lacks a generic model for such traffic. Therefore, similar to [7], in the present work, we assume that content is perfectly replicated.

It is clear from the above discussion that the factors contributing towards transition costs depend not only on how the data center network is deployed and operated but also on the applications being hosted. The utility of modeling a specific deployment is limited. The question that is significant, however, is the impact of variation in the magnitude of transition costs relative to the electricity cost of operation within a given interval on the possible electricity cost savings. Therefore, we have used a normalized and parametrized model for transition costs in our problem formulation so that operators can easily plug-in the parameter values (idling costs, transition costs, number of data centers and their locations) from their own data center deployment.

In this paper, we present Relocate Energy Demand to Better Locations (RED-BL), a framework for optimizing an operator's electricity costs by dynamically re-assigning workload to available data centers at discrete intervals in a planning window. This optimization considers not only the electricity cost of a particular workload assignment, but also the cost of transition from one network state to another.

We find that using RED-BL workload relocation solutions, an operator may save up to 45% of their electric bill, for a wide range of transition costs, compared to the case of uniformly distributing the workload among data centers. On average, RED-BL is 8.28% better as compared to the existing RED-CL solutions [2, 7, 13]. While this percent additional saving may seem modest, it can translate into millions of dollars of annual savings for data center operators. To realize these savings, RED-BL requires a quick computation at the start of each planning interval.

In a short version of this paper [16], we made the following contributions:

1. We proposed RED-BL, the first electric bill minimization framework for

data center operations considering the transition costs.

2. We formulated RED-BL as a network state trajectory optimization problem; the solution (RED-BL) picks a sequence of network states over a *look-ahead* planning window.
3. An evaluation of RED-BL and it's comparison with RED-CL was presented based on trace-driven simulations. Our evaluation used electricity prices from the US markets and workload data from live Internet applications. Our simulations spanned a wide variety of operators (with number of data center varying from 1 to 33), and data centers of varying capacity. We also performed a sensitivity analysis of the RED-BL solution as the cost of activating and deactivating a data center changes.
4. To the best of our knowledge, this is the first study to evaluate the sensitivity of workload relocation solutions (RED-BL and RED-CL) to workload prediction accuracy, amount of over provisioning, and geographical diversity.

In this paper, we extend our work published in [16], and make the following contributions:

1. The electric load for certain type of equipment, such as air-conditioning and networking, may never be curtailed. We term such equipment as inelastic load and the rest of the equipment (only servers, in the present work) as the elastic load. We evaluate RED-BL by modeling the overhead of state changes for elastic load in a data center deployment. In our study, the state change ranges from the (theoretical) extreme of turning servers on and off to the use of DVFS.
2. Workload estimation errors are likely to affect the quality of the RED-BL solution. We propose a sliding-window re-optimization scheme that periodically re-invokes the workload estimation and optimization. We also perform a study of the impact of re-optimization frequency on the amount of savings achievable through RED-BL.
3. Computational resources needed in a given interval depend on the peak workload for that interval. While average workload may be predicted quite accurately, peak workload within an hour is not easily predictable. To schedule resources according to average workload demand and still absorb fluctuations, some reserve resources must be kept active as well. We study the sensitivity of RED-BL's electricity cost savings to the size of reserve computational resources.
4. Granular (de)activation of data center resources is likely to reap more savings in electricity costs. We propose a RED-BL framework that is configurable in the granularity of (de)activation. We study the impact of (de)activation granularity on the amount of electricity cost savings achievable through RED-BL.
5. We show that RED-BL problem is NP-Complete and propose a heuristic algorithm and compare it's performance with the optimal solution using real datasets.

Our solution provides detailed operational planning information in the form of:

- A list of data centers where elastic load must be kept active for each interval in the planning window, and
- The workload distribution amongst these data centers.

The rest of the paper is organized as follows. Section 2 discusses related prior work. Section 3 presents the RED-BL mathematical optimization problem, discusses that the optimal workload mapping problem itself is NP-Complete and provides a heuristic algorithm. Section 4 describes the experimental setup and the data sets that we used. In Section 5 we present the results of our study, followed by a discussion of future directions in Section 6. In Section 7, we discuss the conclusions resulting from our study.

2. Related Work

Li et. al. determined the electricity cost optimal mapping of workload to geo-diverse data centers by controlling the state of the individual servers within each data center [9]. The state of the servers and their electricity consumption was controlled using Dynamic Voltage and Frequency Scaling (DVFS) and Dynamic Cluster Server Configuration (DCSC). Since their optimization problem formulation used Mixed Integer Programming (MIP) with decision variables per server, their approach is effective for small-scale problems such as an individual data center or a fraction thereof. This limitation is evident from the small number of servers used in the simulation-based evaluation in [9]. One way to scale their approach to large distributed data centers is to use coarse granularity in their problem formulation. For instance, instead of controlling the state of each server independently, all servers in a single rack could be configured in the same state at a given time. They used the Soccer World Cup 1998 webserver workload traces and electricity prices at four different locations to evaluate their proposal.

Some researchers have also proposed algorithms for the data center electricity cost optimization problem. In the context of a web-search query processing system hosted on a geo-diverse data center network, Kayaaslan et. al. presented a bin-packing type of algorithm for shifting search query workload between data centers in [17]. Buchbinder et. al. proposed online algorithms for relocating MapReduce jobs between geo-diverse data centers to reduce the electricity bill while considering the cost of inter-data center bandwidth [18]. Their proposed algorithms consider the uncertainty in electricity prices and workload estimates while mapping the jobs to data centers. They evaluated their algorithms using electricity prices from 30 locations across the US and workload data from a 10000 node MapReduce cluster. Bhaskar et. al. proposed online algorithms for mixed packing and covering, a problem which may be applied to optimally map workload to geo-diverse data centers [19]. For configuring servers in a single data center with a view of minimizing electricity costs, Lin et. al. presented offline as well as online algorithms for dynamic scaling of server computational

capacity [10]. Urgaonkar et. al. proposed an online optimization algorithm while proposing to disconnect data center devices from mains and running on UPS when the electricity prices are high [20]. Their proposed scheme recharges the UPS units when the electricity prices are lower.

Investigation of strategies of infrastructure scaling to conserve power in a single data center is reported in [11, 21, 22, 23]. Chen et. al. proposed three different solutions that either shut down or frequency-scale servers in a web hosting data center with the objective of minimizing electricity and maintenance cost while ensuring SLA compliance [11]. The first two of the proposed algorithms were based on a queuing theoretic and control theoretic analysis, respectively, while the third one was a hybrid scheme. While scaling the deployed capacity, their proposed scheme considers the cost of turning the servers on and off in terms of the resulting wear and tear. Mazzucco et. al. present similar strategies in [21]. Oh et. al. considered a virtualized environment and proposed solutions for optimally placing VMs on servers and map workload to the VMs such that electricity costs are minimized [22]. In [23], Chase et. al. present policies for resource allocation in a hosting center alongwith a switching infrastructure for routing requests to servers.

Most of the prior work in this area considers applications with short request-response type jobs. In [24], however, Chen et. al. considered connection-intensive applications such as video streaming, Internet gaming and instant messaging in the context of energy cost aware load dispatch.

Rao et. al. consider data center operation in a futures electricity market and the possibility of hedging against uncertain electricity prices under a smart grid environment in [25]. The authors used workload data from Google search cluster and evaluated a scenario of an operator with data centers at two different locations.

Another related theme of research is greening of data centers. Some examples of work that reports the results of efforts towards *green* mapping of workload to data centers are: [26, 27, 28, 29, 30, 31, 32, 33]. On a related note, Sucevic et. al. studied various approaches for shutting down end-hosts to minimize the total electricity consumption on participant hosts in a peer-to-peer file download system [34].

All of the above work deals with problems that can be categorized broadly as optimal scheduling problems. Such problems arise in many different domains and prior work in such domains is relevant. For instance, System on Chip (SOC) [35], electric power systems and smart grid [36, 37, 38, 39], WiFi access points [40], wide area networks [41], cellular networks [42] and high performance computing [43, 44, 45, 46, 47, 48].

3. Problem Formulation

Some of the electricity load in a data center is inelastic in the sense that it must be on all the time, whereas the rest is elastic in that it may be put into low-power mode (hibernation or standby) or shutdown if there is no workload for

the data center to handle. While formulating the electricity cost minimization problem, we must focus solely on the elastic electricity load in the data center.

According to [49], the breakup of power draw in a typical data center is servers: 56%, cooling: 30%, power conditioning: 8%, network: 5% and lighting: 1%. Cooling often requires a significant ramp up time hence turning on/off cooling may not be a good idea. Power conditioning would be needed to run the inelastic pool of devices. Network equipment may not be turned off due to unpredictable convergence delays whereas lighting load is negligible. In this paper, therefore, we put only the servers in the elastic pool of devices.

Consider a geo-distributed data center infrastructure comprising m interconnected data centers. At any given time, the workload is distributed amongst the data centers in the network. For ease of modeling, we assume that changes to the distribution of workload amongst data centers may be done at the start of discrete intervals of duration λ. We use x_i^j to denote the fraction of workload during interval j that is mapped to data center i. We consider workload that is normalized over it's peak, i.e., the workload values for any interval are between 0 and 1. The workload capacity of data center i, denoted c_i, is also normalized on the same scale. We assume that the network of data centers is over-subscribed so that $\sum_{i=1}^{m} c_i > 1$.

Let the sum of elastic load's peak and idle power consumption over all data centers be P^{max} and P^{min}, respectively. Assuming that the data centers are homogenous, an individual data center's workload capacity is directly related to it's peak (or idle) power consumption. Thus, $P_i^{max} = c_i P^{max}$ and $P_i^{min} = c_i P^{min}$.

Data center power consumption is an affine function of the average CPU utilization of the servers [6]. Therefore, the average power consumption at data center i during interval j is:

$$P_i^j = c_i(P^{min} + \frac{x_i^j(P^{max} - P^{min})}{c_i}) \tag{1}$$

Dividing both sides of the above equation by P^{max} gives the normalized power consumption for data center i during interval j:

$$\hat{P_i^j} = fc_i + x_i^j(1 - f) \tag{2}$$

where f is the ratio of P^{max} to P^{min}. If we set $x_i^j = 0$, i.e., data center i is not computing any workload during interval j, then the second term in equation 2 goes to zero and the power consumption reduces to the first term in equation 2 only, which we refer to as *idle power consumption*. The second term in equation 2 indicates the workload-dependent *computational power consumption*, which is independent of the data center capacity.

Let σ and δ be the average power consumption, over a single interval, required to activate or deactivate, respectively, all of the elastic load at a unit capacity data center. Then, the bootup power consumption for the elastic load at data center i is σc_i. The electricity cost for activating data center i's elastic

load at the start of interval j is, therefore[2], $c_i e_i^j \sigma$. Here, we are assuming that the elastic load at a data center may be activated within a single interval. The value of λ that we used in our experiments is equal to one hour, which is a sufficiently large interval for server activation. Deactivation cost for elastic load may also be derived in a similar manner.

Electricity cost incurred at data center i during interval j is a product of it's total power consumption (computing, idling, activation and deactivation), duration of the interval (λ) and the unit price of electricity (e_i^j). Hence, the RED-BL optimization problem formulation may be given as:

$$\text{minimize} \sum_{j=1}^{n} \sum_{i=1}^{m} c_i e_i^j (p_i^j \lambda (f + (1-f)\frac{x_i^j}{c_i}) + b_i^j \sigma + s_i^j \delta) \tag{3}$$

subject to:

$$x_i^j \leq c_i \quad \forall i, \forall j \tag{4}$$

$$\sum_{i=1}^{m} x_i^j = w^j \quad \forall j \tag{5}$$

$$p_i^j, b_i^j, s_i^j \in \{0, 1\} \quad \forall i, \forall j \tag{6}$$

$$p_i^j \geq x_i^j \quad \forall i, \forall j \tag{7}$$

$$b_i^j \geq p_i^j - p_i^{j-1} \quad \forall i, 2 \leq j \leq n \tag{8}$$

$$s_i^j \geq p_i^{j-1} - p_i^j \quad \forall i, 2 \leq j \leq n \tag{9}$$

$$b_i^0 = p_i^0, s_i^0 = 0 \quad \forall i \tag{10}$$

Decision variable p_i^j is 1 if the elastic load in data center i is active during interval j, or 0 otherwise. Similarly, b_i^j (s_i^j) is 1 if the elastic load in data center i is activated (deactivated) at the start of interval j. In equation (3), multiplication of the first two terms by p_i^j ensures that computation and idling costs are accounted for in interval j, only if the elastic load in data center i is active during that interval. Similarly, multiplication of the last two terms in equation (3) by b_i^j and s_i^j, respectively, ensures that bootup and shutdown costs contribute to the summation only when the elastic load in a data center is booted up or shutdown.

The workload capacity constraint is given in (4). Eq. (5) ensures that all incident workload is handled, while (6) represents the binary-value constraint. Inequality (7) ensures that the elastic load in a data center is active whenever there is any workload mapped to it. The constraint in Eq. (8) ensures that b_i^j is 1 if the elastic load is inactive in interval $j-1$ and active in the next interval. The involvement of b_i^j in the minimization objective function ensures that it is 0

[2]Multiplication with the duration of an interval, i.e., λ is not necessary, because σ is defined as the per interval cost.

Parameter	Description
m	Number of data centers
n	Number of intervals in a planning window
λ	Duration of an interval in hours
f	The ratio between a data center's peak and idle power consumption
c_i	Normalized workload capacity of data center i
σ	Penalty for activating the elastic load at a unit capacity data center as a fraction of its energy consumption at full load in one interval
δ	Penalty for deactivating the elastic load at a unit capacity data center as a fraction of its energy consumption at full load in one interval
e_i^j	Unit cost of electricity at data center i during interval j
w^j	Operator's workload during interval j
x_i^j	Workload mapped to data center i during interval j
p_i^j	1 if data center i is active (either computing workload or idling) during interval j, 0 otherwise
b_i^j	1 if data center i's elastic load is activated at interval j, 0 otherwise
s_i^j	1 if data center i's elastic load is deactivated at interval j, 0 otherwise

Table 1: Data Center Network Model Parameters

otherwise. Similarly, the constraint in Eq. (9) ensures that s_i^j takes on the correct value depending on the deactivation of elastic load in the data centers. We assume that the elastic load in all data centers is initially shutdown, therefore, an activation may be necessary at the first interval whereas deactivation in the first interval is not necessary. These conditions are ensured by the constraints in Eq. (10). It is easy to customize this constraint such that all data centers are assumed to be initially active.

3.1. Problem complexity and a heuristic

The optimal workload relocation problem is identical to the Unit-Commit problem [50] in distributed electricity generation and transmission scenario. In the unit commit problem, one determines the amount of power to be supplied from each generating resource and schedules the activation (ramp up), deactivation (ramp down) and idling (spinning reserves) of the generating resources, given time-varying demand for electricity. Due to a one-to-one mapping between the data center-workload mapping and Unit-Commit problems, it follows that if there is a polynomial time solution for the data center-workload mapping problem, Unit-Commit may also be solved in polynomial time. However, since the Unit-Commit problem is known to be NP-Complete [50], it follows that so is the workload-mapping problem that we are considering in this paper. We will show later that we are able to solve reasonably large sized instances of the above NP-Hard MIP formulation for RED-BL using the CPLEX solver. Nonetheless, we now present a heuristic algorithm for it. The overall worst case

running time[3] of the heuristic algorithm, given in in Algorithm 1, is O(mn^2 + n^3 + $mlgm$).

The pseudo-code of our heuristic algorithm for RED-BL is given in Algorithm 1. Assume that the workload vector for the planning window starts at a trough, then rises in a non-decreasing manner to the peak before falling off in a non-increasing manner to another trough. Since the activation/deactivation costs are expected to be significant, our heuristic is designed to select a small number of data centers to operate in long continuous stretches during a given day. For the assumed characterization of the workload, elastic load at a few data centers would be sufficient to handle the workload early (and late) in the planning window. As the workload rises gradually, elastic load at some more data centers would need to be brought online. As the workload starts to fall, elastic load at some data centers may gradually be deactivated until the planning window ends. Our heuristic places two pointers at the beginning and end of the planning window, determines the number of data centers (d_1 and d_2) needed to handle the workload corresponding to the two pointers and picks the smaller of these two values. It then finds min(d_1, d_2) best data centers in terms of having the least average electricity price over the planning window. The elastic load at these data centers will be kept active between the intervals corresponding to the two pointers. Furthermore, our algorithm assigns as much workload as possible to the selected data centers in ascending order of average electricity price in the chosen intervals.

As long as the workload in the intervals corresponding to the two pointers may be handled by the same number of data centers, both of the pointers are moved closer to each other. Otherwise, the pointer that corresponds to the interval requiring the smaller number of data centers is moved towards the other pointer. This pointer movement is performed until either the pointers cross each other or the number of data centers required to handle the workload in the interval corresponding to the moving pointer increases. In the former case, we are done and in the latter, the algorithm repeats the data center selection and workload mapping step. The algorithm then activates elastic load at data center(s) to meet the workload requirement of the pointer corresponding to the interval with the higher workload. The data center(s) where the elastic load is activated at this time are the ones that are not used before and have the least average electricity price in the interval between the two pointers.

4. Experimental setup

In this section, we describe the experimental setup to perform a comparative study of different workload placement algorithms under various scenarios.

[3]To conserve space, we have omitted the detailed but straightforward derivation of the running time of our heuristic.

Algorithm 1 Heuristic for the RED-BL problem

Require: $w[1..n]$: Cumulative data center workload for the planning window,
$e[1..m][1..n]$: Electricity prices for all data centers over the planning window
Ensure: $z[1..m][1..n]$: workload assigned to the data centers for all intervals
$y[1..m][1..n]$: Data center status (1=on/0=off) over the planning window

$g_1 = 0; \quad g_2 = n - 1; \quad l = w; \quad a = 1..m; \quad n_c = 0;$
  $y[i][j] = 0; \quad z[i][j] = 0; \quad (\forall i, \quad \forall j)$
repeat
  $d_1 = \lceil w[g_1]/c_1 \rceil; \quad d_2 = \lceil w[g_2]/c_1 \rceil; \quad n_d = \min(d_1, d_2)$
  if $n_d > n_c$ **then**
    Sort a in ascending order of average electricity price in $[g_1, g_2]$
    for all $i \in a$ **do**
      for all $j \in [g_1, g_2]$ **do**
        $y[i][j] = 1; \quad n_c{+}{+}$
        $z[i][j] = \min(l[j], c_i)$
        $l[j] = l[j] - z[i][j]$
        Remove i from a
      end for
    end for
  end if
  repeat
    $g_1{+}{+}$
  until $(\lceil w[g_1]/c_1 \rceil > n_c)$**or**$(g_1 > g_2)$**or**$(\lceil w[g_1]/c_1 \rceil > \lceil w[g_2]/c_1 \rceil)$
  repeat
    g_2- -
  until $(\lceil w[g_2]/c_1 \rceil > n_c)$**or**$(g_1 > g_2)$**or**$(\lceil w[g_1]/c_1 \rceil < \lceil w[g_2]/c_1 \rceil)$
until $g_1 > g_2$

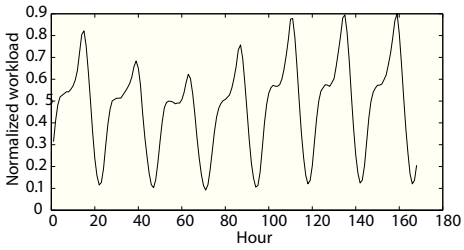

Figure 2: Normalized workload

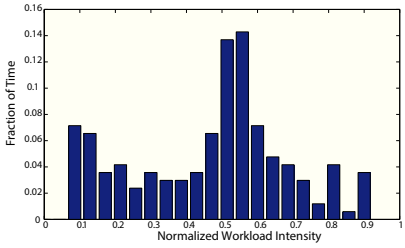

Figure 3: Workload intensity histogram

4.1. Application workload

We used an year-long trace of hourly workload for 3 social networking applications, with a subscription base of over 8 million users [51]. In order to make the dataset representative of a large data center network operator, we aggregated these traces into a week long trace as follows. We sliced the trace into week-long segments and considered each slice as workload for a different application, for the same week. We, then, normalized the sum of these trace vectors so that the peak cumulative workload corresponds to a value of 0.9. The normalized workload intensity is plotted in Figure 2. The statistical characteristics of our workload, as plotted in Figure 3 are quite similar to those reported by Google for "thousands of servers during a six-month interval at a Google data center" [5].

4.2. Electricity prices

We selected 33 different regions in the USA for which hourly electricity prices are available online. These regions belong to the following Independent System Operators (ISOs): NYISO, CAISO, MISO, ISO-NE and PJM. We used the day-ahead prices for these locations, i.e., the electricity price negotiated for the same hour on the following day. In all the experiments for this work, we considered an operator with data centers at all 33 locations in our dataset.

4.3. Algorithms for Workload Distribution/Relocation

The workload relocation problem has the following dimensions based on which different algorithms may be formulated.

- For a given interval, the strategy for distribution of workload amongst data centers.
- For a given interval, the strategy for the state (on/off) of elastic load at a data center which has not been assigned any workload. In such cases, there is a trade-off between keeping the elastic load on (and incurring idling costs) and deactivating it (while incurring deactivation overhead and possibly activation overhead if it needs to be brought back online later in the planning window).

- Over the planning window, does the algorithm report transition costs in the total electricity cost?

In this paper, we report comparative results for six workload placement algorithms.The following list describes and differentiates these algorithms. The same comparison is also presented in tabular form in Table 2.

- **RED-BL:** This is our proposed algorithm that determines the global optimal cost of electricity over a planning window while considering and reporting the transition costs. The choice of workload distribution as well as the state of elastic resources with no workload is governed by the optimal solution as determined by the CPLEX solver.
- **Heuristic:** This is the heuristic algorithm that we proposed in Section 3.
- **UNIFORM:** This algorithm represents the choice of those operators that find an even loading of their data centers desirable. This algorithm does not deactivate elastic loads and hence does not incur transition costs.
- **Greedy algorithms:** The originally proposed algorithm in [7] distributes workload to data centers such that, for each interval in the planning window, it makes a greedy assignment (in terms of current electricity price) of workload to data centers. Furthermore, this original algorithm keeps the elastic load at all data centers active in all intervals, incurring significant idling costs and hence is naturally disadvantaged against RED-BL. To have a fair comparison with the greedy workload distribution strategy, we use several variants of the original algorithm as well.
  - **Local optimal with Idling (LI):** This is the originally proposed algorithm from [7]. It does not deactivate elastic load.
  - **Local optimal withOut transition costs (LO):** This variant of LI was proposed in [7]. It deactivates un-needed elastic load while ignoring the transition costs. This algorithm does not report transition costs in the total electricity cost of it's proposed workload mapping for the planning window. This algorithm is very useful because it defines the lower bound on electricity cost that any algorithm can ever achieve.
  - **Local optimal with Deactivation (LD):** This algorithm is similar to LO in all respects except that it also reports the activation/deactivation costs as part of the total cost of it's proposed solution. Unlike LO, it's results are practically relevant. It's total cost is less than (for all practical cases) LI, which makes it somewhat competitive to RED-BL.
  - **Local optimal with Selection (LS):** In cases where transition costs are high compared to idling costs it would be better to keep the elastic resources at a data center active and incur idling costs if it will be needed again after the lapse of a small number of intervals.

Workload mapping strategy	
LI, LD, LS, LO	Greedy
RED-BL	Based on global optimal solution
UNIFORM	Workload equally divided amongst all data centers
State of a data center in an interval when it has no workload	
LI	Active and idling
LD, LO	Inactive
LS	Either inactive or idling, whichever is cheaper
RED-BL	Based on global optimal solution
UNIFORM	Active
Is transition cost reported in the total electricity cost reported?	
LI	N/A
LD, LS	Yes
LO	No
RED-BL	Yes
UNIFORM	N/A

Table 2: A comparison of the algorithms studied in this paper

LS is a variant of LD that is empowered with the ability to *select* whether to deactivate unneeded elastic load at a data center or keep it idling. The cost of LS is never greater than that of LD.

5. Results

To evaluate the utility of workload relocation for electricity cost minimization, we formulated seven different scenarios. For each scenario, we ran seven experiments (one for each day's workload in our dataset) and report the average of the total electricity cost for each algorithm. Each experiment determines an operational plan for a planning window consisting of 24 consecutive intervals, each with a duration of one-hour.

5.1. Scenario 1 (Extent of over provisioning)

In this scenario, we investigate the relationship of the amount of data center capacity over-provisioning with the electricity cost savings. As we increase the amount of over-provisioning, each individual data center's capacity would increase enabling more and more workload to be mapped to data centers at locations with cheaper electricity price.

With data centers at all 33 locations in our dataset, we varied c_i between 0.03 and 0.12 (in increments of 0.01). This covers a variety of operators whose

workload capacity ranges from just over expected peak workload to almost 300% over-provisioning.

We computed the total electricity cost for all algorithms while setting $f = \sigma = \delta = 0.65$. The percentage savings in total electricity cost by various algorithms compared to UNIFORM are plotted against the data center capacity over-provisioning in Figure 4. We found that for the wide range of capacity over-provisioning that we considered, LI is able to do only slightly better than the naive UNIFORM algorithm (about 2%). This is due to the significant idling costs incurred by LI under the experiment's conditions. For this reason, we have omitted LI from this plot.

The most competitive practical variants of LI, i.e., LS was 10.35% off from the ideal lower bound (LO). Meanwhile, RED-BL solution is quite close the ideal lower bound (LO). The reason for greater savings with RED-BL compared to the greedy solutions (LS and LD) is that, the transition costs being significant, the former does fewer state transitions. In several intervals, RED-BL chooses data centers with relatively higher electricity price than the greedy solutions, but makes up for the higher computational cost by a reduction in the transition costs incurred.

5.2. Scenario 2 (Activation/Deactivation overhead)

As the magnitude of transition costs relative to the state cost for an interval grows beyond a certain point, the benefits of deactivating elastic load at data centers would diminish. Accordingly, the electricity cost savings achievable by the workload relocation schemes would drop with increase in transition costs. In this scenario, we determine the percentage savings in total electricity cost for each algorithm compared to UNIFORM, while varying the activation/deactivation overhead between 0 and 1, in increments of

Since LI does not (de)activate unneeded elastic load, it's electricity cost is independent of the magnitude of transition costs. We observed that it offered a saving of merely 1.74% compared to UNIFORM. Figure 5 shows the electricity cost savings for the other algorithms compared to UNIFORM. The LS and LD variants offered savings that scale almost linearly to the magnitude of transition costs. Compared to LI, both LS and LD bring a factor of 4 reduction in the electricity cost, on average. RED-BL not only scales better than LS and LD but also achieves electricity cost saving that is fairly close to the ideal lower bound as reported by LO (only 3% higher, on average).

5.3. Scenario 3 (Granular activation/deactivation)

In this scenario, we investigate the potential benefits of (de)activating fixed size subsets of the elastic load in a data centers instead of an *all or nothing* (de)activation approach. Granular (de)activation is expected to bring additional power savings. For instance, if we are only allowed to (de)activate the entire elastic load in a data center that is operating at 10% of it's capacity, then 90% of it's elastic load is still consuming significant idling costs. However, if we had the ability to deactivate half of the elastic load at a data center, we could cut idling energy cost significantly.

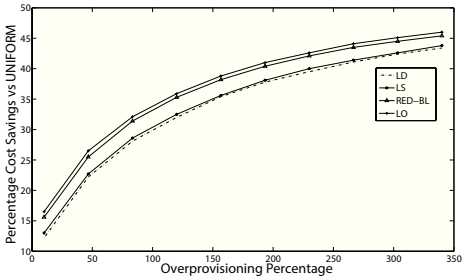

Figure 4: Percentage electricity cost savings vs over-provisioning (compared to UNIFORM)

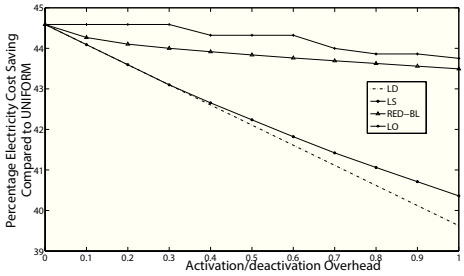

Figure 5: Percentage electricity cost savings vs transition overhead (compared to UNIFORM)

The size of the portion of the elastic load in a data center that may be independently (de)activated may be deployment-dependent or operator-dependent. Possible choices of granularity may be a rack, a pod or one half of the elastic load etc. The RED-BL optimization problem formulation with l granular (de)activation levels is given by:

$$\text{minimize } \sum_{j=1}^{n}\sum_{i=1}^{m} c_i e_i^j (p_i^j \lambda (\frac{f}{l} + (1-f)\frac{x_i^j}{c_i}) + \frac{b_i^j \sigma}{l} + \frac{s_i^j \delta}{l})$$

subject to:

$$x_i^j \leq c_i \quad \forall i, \forall j \tag{11}$$

$$\sum_{i=1}^{m} x_i^j = w^j \quad \forall j \tag{12}$$

$$p_i^j, b_i^j, s_i^j \in \{0, 1, ..., l\} \quad \forall i, \forall j \tag{13}$$

$$p_i^j \geq x_i^j * l / c_i \quad \forall i, \forall j \tag{14}$$

$$b_i^j \geq p_i^j - p_i^{j-1} \quad \forall i, 2 \leq j \leq n \tag{15}$$

$$s_i^j \geq p_i^{j-1} - p_i^j \quad \forall i, 2 \leq j \leq n \tag{16}$$

$$b_i^0 = p_i^0, s_i^0 = 0 \quad \forall i \tag{17}$$

There are three primary differences from the vanilla RED-BL formulation. The first difference is in the objective function, where the idling, bootup and shutdown costs depend on the number of granular units involved in the idling, bootup or shutdown process respectively. The second difference is in the domain of p_i^j, b_i^j and s_i^j (see constraint (13)). The third difference is in the constraint (14), which ensures that p_i^j takes on an appropriate value from $0, 1, ..., l$. Since computational cost is independent of the data center capacity, it is also independent of the number of granular units being used at a data center during a given interval.

In Figure 13, we have plotted the percentage savings in electricity cost vs the granularity of data center's elastic load (de)activation. The savings are compared against the scenario where the entire elastic load in the data center may only be (de)activated as a whole ($l = 1$). Accordingly, in Figure 13 we see no savings for $l = 1$. We also see that the ability to independently (de)activate half of a data center's elastic load provides around 2.5% additional electricity cost savings on top of what vanilla RED-BL can achieve. The electricity cost savings grow almost linearly when going to more granular size of independent (de)activation.

5.4. Scenario 4 (DVFS)

Dynamic Voltage and Frequency Scaling (DVFS) offers an alternative to shutting down servers when they are not needed. We evaluate the power savings achievable by RED-BL versus the power-reduction possible through DVFS. According to [52], DVFS may be used to reduce server power consumption to 18% of it's peak power consumption. In order to keep our results generic, we have experimented with all possible values of power consumption reduction factor for DVFS in conjunction with our dataset. The values of f, σ and δ were fixed at 0.65 in these experiments and the results are plotted in Figure 6. Our results suggest that under the experimental conditions, if servers at un-utilized data centers were low-powered using DVFS, we could save about 60% on electricity costs compared to the case where such data centers were allowed to run on idle. The scaling of power cost reduction is linear and tapers out when idling power consumption is less than the power consumption offered by DVFS.

5.5. Scenario 5 (Margin for short-term workload variation)

We have used traces of cumulative hourly workload, which is quite smoothed out, whereas there are variations in instantaneous workload on short time-scales. If capacity is provisioned for average workload, then queues would build up sometimes with a negative impact on performance. In order to avoid that, some reserve margin in data center capacity may be kept in every interval on top of the expected workload.

Figure 7 shows the relationship between the reserve margin and the electricity cost. The reserve margin is varied between 1% and 10% of a data center's capacity. The base-line is RED-BL cost when no margin is kept, and workload is well-behaved. The lower line in the graph shows the percentage increase from the baseline if the actual average workload is equal to what was estimated, i.e., the reserve capacity did not compute any workload and only accounted for idling costs. The upper line in the graph shows the case if workload is so excessive that all reserve capacity is needed to compute workload. Depending upon the magnitude of the spike above the expected workload, the actual difference in electricity cost compared to the baseline would be somewhere between the two lines.

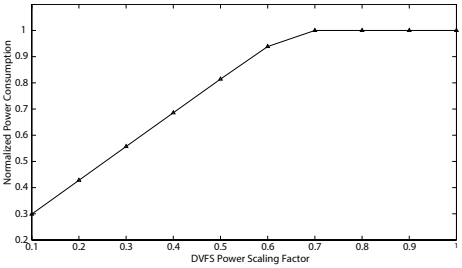

Figure 6: If DVFS is used to reduce server power consumption instead of shutting them down, the electricity cost savings would be reduced, but the approach would be more acceptable to risk-averse operators. Here, we plot the difference in RED-BL electricity cost when DVFS is used to reduce server power consumption between 0% (complete shutdown) and 100% (unused servers run on idle).

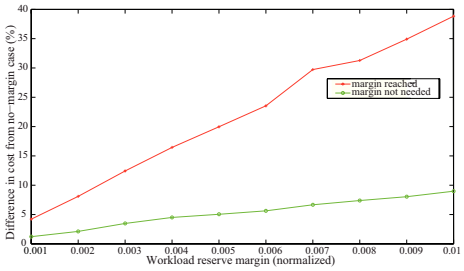

Figure 7: Some reserve margin of server capacity must be reserved at each data center to handle unexpected workload variations. This figure plots the percentage increase in RED-BL electricity cost for varying degrees of reserve margin compared to the RED-BL electricity cost in the absence of reserve margin and well-behaved workload. The upper line shows the worst-case, i.e., workload exhausts all reserves. The lower line is the best-case, i.e., reserve capacity is not needed.

5.6. Scenario 6 (Sliding Window Optimization)

All of our prior simulation scenarios were driven by error-free workload traces. The underpinning assumption to the corresponding results, therefore, is the availability of accurate workload estimates. We opine that this is not such a bad assumption given that the cumulative workload on the granularity of an hour changes slowly from one hour to the next and from one hour on a day to the same hour the next day. Nevertheless, workload forecasting will have some error, however small.

The state trajectory proposed by RED-BL for a planning window would be different if erroneous workload estimates were used instead of error-free estimates. In particular, the projected state in a given interval may be *infeasible* in the sense that the active resources may be insufficient for the actual workload. In such a case, some more resources may need to be activated. Similarly, the projected state for some interval may be *sub-optimal* in the sense that more resources may be active than what is needed for the actual workload. In such a case, it may be desirable to deactivate some of the resources. In a real deployment, therefore, one must periodically correct the projected state trajectory to be as close to the optimal state trajectory as possible.

Receding horizon control (RHC) [53] is a strategy that is commonly used in such situations. At a given interval, RHC makes a forecast for a number of future intervals, known as the horizon. Based on the projected horizon, the optimal current state is picked. The forecasting and state correction is repeated at every interval to get more accurate forecasts for future intervals by exploiting the availability of more historic data about the workload. We call this step of repeated forecasting and subsequent generation of a RED-BL plan *global trajectory correction*.

Since the RED-BL optimization problem is an NP-Complete problem, it may not be feasible for very large scale deployments to invoke it at every interval. For this reason, we formulated a generalization of the RHC which we call sliding window re-optimization. We define a parameter γ called the window slide latency. At interval number 1, the workload for the next n intervals is forecast[4] and a projected RED-BL state trajectory is calculated. The same thing is done γ intervals later. In other words, the planning over an n interval horizon is done at a longer time-scale compared to RHC. To accommodate for infeasible or sub-optimal states in the intervals 1 through γ, i.e., on a short time-scale, we perform a *local trajectory correction*. The local trajectory correction only looks at the projected state in the current interval and the actual workload to determine a corrected state for the current interval only. This avoids computation of states over an entire planning window reducing the size of the NP-Complete problem that needs to be solved.

The local trajectory correction step is shown in Figure 11. We start at the initial state S_0. Based on workload forecast for the next n intervals, we project a RED-BL state trajectory and transition to the projected state $\hat{S}_1$ at the beginning of interval 1. However, some time during interval 1, we realize that the actual workload is different from our estimates. To adapt to this situation, we transition to a locally better state S_1. Until γ intervals elapse, we will continue to perform local trajectory correction. That is, we will assume that the estimate for interval 2 is accurate and transition to the planned state $\hat{S}_2$. During interval 2, the local trajectory correction process repeats once we realize that the actual workload is different from the estimated one.

The local trajectory correction for interval j is an optimization problem that attempts to minimize the electricity cost of the corrected state S_j and the cost of transition between the planned state $\hat{S}_j$ and the corrected state. The mixed integer linear programming formulation for the local trajectory correction step for interval j is as follows:

$$\text{minimize} \sum_{i=1}^{m} c_i e_i^j (p_i^j \lambda (f + (1-f)\frac{x_i^j}{c_i}) + (b_i^j + \hat{b}_i^j)\sigma + (s_i^j + \hat{s}_i^j)\delta) \tag{18}$$

subject to:

$$x_i^j \leq c_i \quad \forall i \tag{19}$$

$$\sum_{i=1}^{m} x_i^j = w^j \tag{20}$$

$$p_i^j, b_i^j, s_i^j, \hat{p}_i^j, \hat{b}_i^j, \hat{s}_i^j \in \{0, 1\} \quad \forall i \tag{21}$$

[4]For workload forecasting, we trained an ARMA(4, 4) [54] model on a day's workload and used it to forecast the workload for the rest of the week.

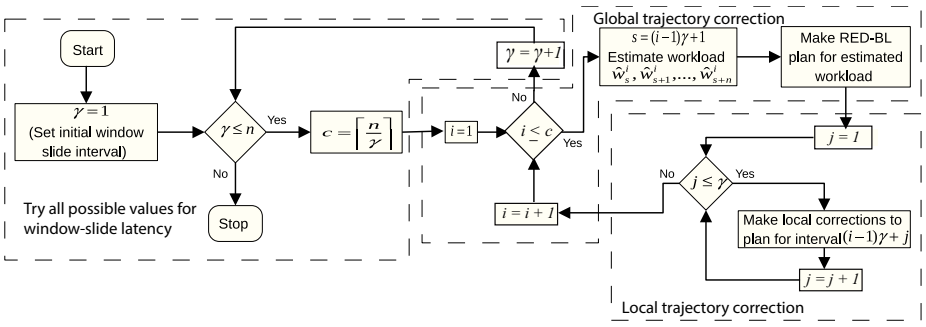

Figure 8: Flow for Sliding Window Optimization Experiments

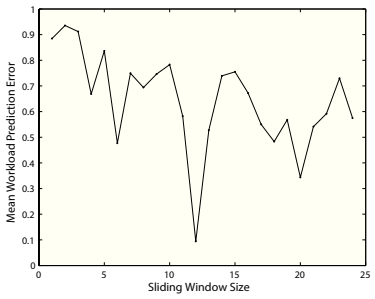

Figure 9: Mean absolute workload prediction error vs sliding window size

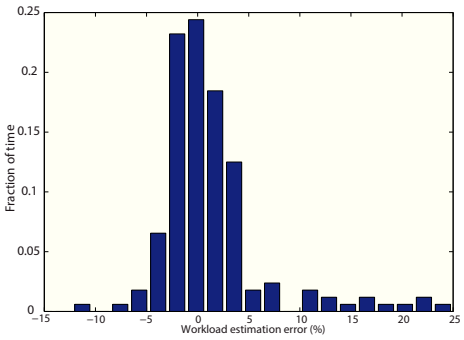

Figure 10: Distribution of workload prediction error for sliding window size of 12 hours

$$p_i^j \geq x_i^j \quad \forall i \tag{22}$$

$$b_i^j \geq p_i^j - \hat{p}_i^j \quad \forall i \tag{23}$$

$$s_i^j \geq \hat{p}_i^j - p_i^j \quad \forall i \tag{24}$$

$$\hat{b}_i^j \geq \hat{p}_i^j - p_i^{j-1} \quad \forall i \tag{25}$$

$$\hat{s}_i^j \geq p_i^{j-1} - \hat{p}_i^j \quad \forall i \tag{26}$$

Given that the planning window size is n intervals, the possible values for γ are $2, 3, .., \gamma$. Figure 8 shows the flow of our experiments. The leftmost dashed polygon represents the loop that cycles through all possible values of γ. As an example, consider the iteration of the outer loop where $\gamma = 2$. We start by forecasting the workload for the first n-intervals, denoted by $\hat{W}_1^2 = [\hat{w}_{1,1}^2, \hat{w}_{2,1}^2, ..., \hat{w}_{n,1}^2]$. Here, $\hat{w}_{j,1}^2$, for instance, represents the workload forecast for interval j during the first forecasting operation while the value of γ is 2. Using $\hat{W}_1^2$ as the expected workload vector, we propose a RED-BL deployment plan for the first n-intervals. At the start of the third interval (after the lapse of γ intervals), we forecast the workload for the next n intervals, leveraging the additional information about the actual workload for the first two intervals which was not available in the first

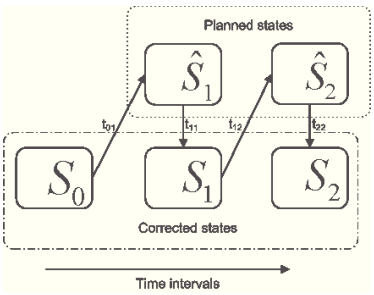

Figure 11: Local trajectory correction technique for three consecutive intervals

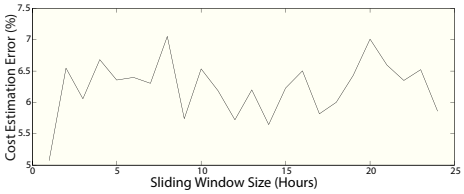

Figure 12: Percentage error of sliding window forecasts compared to global optimal with error-free workload

forecast step at $t = 0$. This forecast is denoted by $\hat{W}_2^2 = [\hat{w}_{3,2}^2, \hat{w}_{4,2}^2, ..., \hat{w}_{n+2,2}^2]$. Then, we compute the RED-BL deployment plan for intervals $3, 4, ..., n + 2$ as the global trajectory correction step. Since the window sliding interval size is γ and the number of intervals in our experiments is n, the number of times the window must slide, for a given value of γ is $\lceil n/\gamma \rceil$.

Having trained an ARMA(4,4) model on the first day's data, we ran experiments for the last six days' workload in our dataset. We computed the average error of the daily electricity cost reported by these experiments compared to the total daily electricity cost for the same period with perfect workload estimates. The size of the planning window was set to 24 hours.

The first set of results in this scenario is the percentage workload estimation error for various sliding window sizes. We see in Figure 9 that the mean absolute percentage prediction error is less than 1%. The minimum mean error is for a sliding window size of 12 hours. For this sliding window size, the distribution of percentage workload estimation error is plotted in Figure 10. Most of the workload estimates are quite close to error-free, while a few estimates are as much as 24% off. This low average error for $\gamma = 12$ is expected due to the diurnal cycles in workload volume.

The difference of the electricity cost resulting from the use of the sliding window trajectory correction approach compared to the optimal solution with perfect workload knowledge is plotted in Figure 12. We see that the electricity cost achievable with RED-BL in a sliding window fashion is within 5-7% of the optimal cost achievable with perfect workload estimates for all values of γ. Also note that the pure RHC strategy turns out to be the best. This is expected because a RED-BL plan based on knowledge of the current actual workload as well as forecast for the next $n-1$ intervals is better than a greedy decision based only on the current actual workload.

5.7. Scenario 7 (Performance of the Heuristic Algorithm)

Figure 14 shows the performance of the heuristic algorithm that we proposed in Section 3 compared to the optimal solution of the problem for various values of the (de)activation overhead parameters. For each value of the b and s

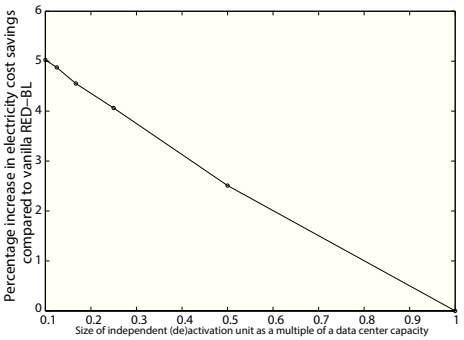

Figure 13: Cost saving vs (de)activation granularity

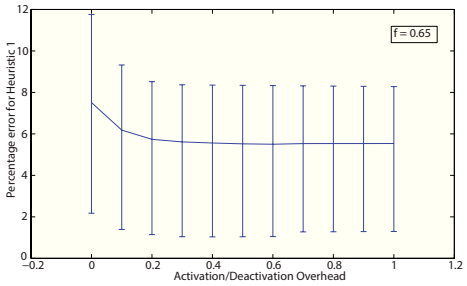

Figure 14: The minimum, maximum and average percentage difference between the cost of our heuristic and RED-BL

parameters, we have plotted the average error over the seven days in our workload dataset (the curve) as well as the minimum and maximum error for any given day (the vertical bars). Since our heuristic is designed to avoid activation/deactivation, it performs poorly when the value of b and s parameter is low. As the value of b and s increases, our heuristic's error compared to the optimal solution drops until it starts a slight rise. When the value of b and s parameters is higher compared to the value of f, it may sometimes be better to activate the elastic resources in a data center a few intervals earlier than they are needed. Suppose that the resources at a certain data center are needed in interval i. There may be some interval $i-\epsilon$ that has a lower electricity price and the sum of the idling costs for intervals $i-\epsilon$ through $i-1$ plus the activation cost in interval $i-\epsilon$ may be less than the cost of activation in interval i. It is also possible that some times delayed deactivation of elastic resources may be better in the long rum. We observed similar trends for other values of f as well, when b and s parameter values are varied from 0 to 1.

6. Discussion

Our work opens several avenues for further studies. Some of these future directions involve considering more specific sources of transition costs instead of an abstract model of transition costs.

- Deactivating the elastic load at data centers with no-load might change the latency to some of the clients. Increased latency is reported to result in loss in revenue [55]. This information could be incorporated into the RED-BL optimization problem to maximize the operator revenue instead of simply minimizing the electricity cost.

- Due to convergence delays inherent in the relocation mechanism, some clients might notice the change in workload mapping only after the lapse of considerable delay following a network state change. Meanwhile, the operator can not deactivate the "old" data centers because some workload

would continue to be routed there. This poses additional challenges for the RED-BL framework and a more detailed study of the trade-offs between energy cost and performance would be useful.

- Inter-data center traffic costs are quite high [1]. Maintaining replication amongst data centers will incur some overhead in terms of the cost of replication traffic. Furthermore, if the elastic load at a data center is re-activated after being inactive for several intervals, it is unclear how much cost would be incurred to bring the replica back to the same level of consistency as the rest of the network. We think that it would be useful to investigate how this can be incorporated into the transition costs.

7. Conclusion

Geo-temporal diversity in electricity prices coupled with geographic diversity in typically over-provisioned data center network suggests a possibility of smartly allocating resources to save electricity costs. Previously proposed approaches to such dynamic workload relocation mostly ignored the cost of transitions between data center network states in consecutive intervals. We have provided an extensive simulation study of this idea while considering such transition costs.

Our results indicate that ignoring transition costs can result in a significant erosion of possible electricity cost savings. Furthermore, our approach of incorporating transition costs in a global optimization, called RED-BL, scales better with the magnitude of transition costs than the previously proposed RED-CL approaches.

Frameworks such as ours rely on accurate estimates of workload. To compensate for errors in workload estimates, we propose RED-BL with trajectory correction, whereby we perform a revision of workload estimates on a sliding-window basis, coupled with a local re-optimization of the network state at every interval in a planning window. Through our experiments, we found that using workload predicted by an ARMA model, RED-BL with trajectory correction achieves an electricity cost that is, on average, around 6% of the cost achievable using perfect workload estimates.

In another set of experiments, we found that if an operator is able to (de)activate portions of the elastic load in their data center, they can use RED-BL to cut their electricity cost even further. The additional savings increase almost linearly as the number of units of (de)activation within a data center increases.

Bibliography

[1] A. Greenberg, J. Hamilton, D. A. Maltz, P. Patel, The cost of a cloud: Research problems in data center networks, Computer Communications Review 39 (1) (2009) 68–73.

[2] A. Qureshi, Plugging Into Energy Market Diversity, in: 7th ACM HotNets, Calgary, Canada, 2008, pp. 1–6.

[3] K. G. Brill, The invisible crisis in the data center: The economic meltdown of moore's law, Tech. rep., The Uptime Institute (2007).

[4] C. L. Belady, In the data center, power and cooling costs more than the IT equipment it supports, Electronics Cooling 23 (1).

[5] L. A. Barroso, U. Holzle, The case for energy-proportional computing, Computer 40 (2007) 33–37.

[6] X. Fan, W.-D. Weber, L. A. Barroso, Power provisioning for a warehouse-sized computer, in: ISCA, ACM Press, New York, NY, USA, 2007, pp. 13–23.

[7] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, B. Maggs, Cutting the Electric Bill for Internet-Scale Systems, in: ACM SIGCOMM, Barcelona, Spain, 2009, pp. 123–134.

[8] D. Meisner, B. T. Gold, T. F. Wenisch, Powernap: eliminating server idle power, in: Proceedings of the 14th international conference on Architectural support for programming languages and operating systems, ASPLOS XIV, ACM, New York, NY, USA, 2009, pp. 205–216. doi:10.1145/1508244.1508269.
URL http://doi.acm.org/10.1145/1508244.1508269

[9] J. Li, Z. Li, K. Ren, X. Liu, Towards optimal electric demand management for internet data centers, IEEE Trans. Smart Grid 3 (1) (2012) 183–192.

[10] M. Lin, A. Wierman, L. Andrew, E. Thereska, Dynamic right-sizing for power-proportional data centers, in: INFOCOM, 2011, pp. 1098 –1106.

[11] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, N. Gautam, Managing server energy and operational costs in hosting centers, in: ACM SIGMETRICS, ACM, New York, NY, USA, 2005, pp. 303–314.

[12] M. Mazzucco, D. Dyachuk, Balancing electricity bill and performance in server farms with setup costs, Future Generation Computer Systems 28 (2) (2012) 415 – 426.

[13] L. Rao, X. Liu, L. Xie, W. Liu, Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-Electricity-Market Environment, in: INFOCOM, San Diego, USA, 2010, pp. 1–9.

[14] J. Pang, A. Akella, A. Shaikh, B. Krishnamurthy, S. Seshan, On the responsiveness of DNS-based netowrk control, in: IMC, ACM, 2004, pp. 21–26.

[15] C. Labovitz, A. Ahuja, A. Bose, F. Jahanian, Delayed Internet Routing Convergence, in: Proc. of the ACM SIGCOMM, 2000, pp. 175–187.

[16] M. S. Ilyas, S. Raza, C.-C. Chen, Z. A. Uzmi, C.-N. Chuah, RED-BL: Energy solution for loading data centers, in: INFOCOM'12, 2012, pp. 2866–2870.

[17] E. Kayaaslan, B. B. Cambazoglu, R. Blanco, F. P. Junqueira, C. Aykanat, Energy-price-driven query processing in multi-center web search engines, in: 34th ACM SIGIR, ACM, New York, NY, USA, 2011, pp. 983–992.

[18] N. Buchbinder, N. Jain, I. Menache, Online job-migration for reducing the electricity bill in the cloud, in: Proceedings of the 10th international IFIP TC 6 conference on Networking - Volume Part I, NETWORKING'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 172–185.

[19] U. Bhaskar, L. Fleischer, Online mixed packing and covering, CoRR abs/1203.6695.

[20] R. Urgaonkar, B. Urgaonkar, M. J. Neely, A. Sivasubramaniam, Optimal power cost management using stored energy in data centers, in: SIGMETRICS, ACM, New York, NY, USA, 2011, pp. 221–232.

[21] M. Mazzucco, D. Dyachuk, R. Deters, Maximizing cloud providers revenues via energy aware allocation policies, CoRR abs/1102.3058.

[22] F. Y.-K. Oh, H. S. Kim, H. Eom, H. Y. Yeom, Enabling consolidation and scaling down to provide power management for cloud computing, USENIX HotCloud, USENIX Association, Berkeley, CA, USA, 2011, pp. 14–14.

[23] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, R. P. Doyle, Managing energy and server resources in hosting centers, SIGOPS Oper. Syst. Rev. 35 (5) (2001) 103–116.

[24] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, F. Zhao, Energy-aware server provisioning and load dispatching for connection-intensive internet services, in: USENIX NSDI, USENIX Association, Berkeley, CA, USA, 2008, pp. 337–350.

[25] L. Rao, X. Liu, L. Xie, Z. Pang, Hedging against uncertainty: A tale of internet data center operations under smart grid environment, Smart Grid, IEEE Transactions on 2 (3) (2011) 555 –563.

[26] K. Le, R. Bianchini, M. Martonosi, T. D. Nguyen, Cost- and energy-aware load distribution across data centers, in: HotPower, 2009, pp. 1–5.

[27] X. Zheng, Y. Cai, Energy-aware load dispatching in geographically located internet data centers, Sustainable Computing: Informatics and Systems 1 (4) (2011) 275 – 285.

[28] G. Koutitas, P. Demestichas, Challenges for energy efficiency in local and regional data centers, Journal of Green Engineering 1 (1) (2010) 1 – 32.

[29] Z. Abbasi, T. Mukherjee, G. Varsamopoulos, S. Gupta, Dahm: A green and dynamic web application hosting manager across geographically distributed data centers12, Atlanta 60 (2011) 80.

[30] L. Chiaraviglio, I. Matta, Greencoop: cooperative green routing with energy-efficient servers, in: Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, ACM, 2010, pp. 191–194.

[31] D. H. Phan, J. Suzuki, R. Carroll, S. Balasubramaniam, W. Donnelly, D. Botvich, Evolutionary multiobjective optimization for green clouds, in: ACM GECCO 2012, ACM, 2012, pp. 19–26.

[32] Z. Liu, M. Lin, A. Wierman, S. H. Low, L. Andrew, Greening geographical load balancing, in: ACM SIGMETRICS, ACM, San Jose, California, USA, 2011, pp. 233–244.

[33] Z. Liu, M. Lin, A. Wierman, S. H. Low, L. Andrew, Geographical load balancing with renewables, in: ACM GREENMETRICS, ACM, 2011, pp. 62–66.

[34] A. Sucevic, L. L. Andrew, T. T. Nguyen, Powering down for energy efficient peer-to-peer file distribution, SIGMETRICS Perform. Eval. Rev. 39 (3) (2011) 72–76.

[35] Z. Fang, L. Zhao, R. R. Iyer, C. F. Fajardo, G. F. Garcia, S. E. Lee, B. Li, S. R. King, X. Jiang, S. Makineni, Cost-effectively offering private buffers in SoCs and CMPs, in: Proceedings of the international conference on Supercomputing, ICS '11, ACM, New York, NY, USA, 2011, pp. 275–284.

[36] F. Javed, N. Arshad, On the use of linear programming in optimizing energy costs, in: 3rd IWSOS, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 305–310.

[37] T. Logenthiran, D. Srinivasan, A. M. Khambadkone, Multi-agent system for energy resource scheduling of integrated microgrids in a distributed system, Electric Power Systems Research 81 (1) (2011) 138 – 148.

[38] G. Celli, F. Pilo, Optimal distributed generation allocation in mv distribution networks, in: 22nd IEEE PICA 2001, 2001, pp. 81 –86.

[39] F. Javed, N. Arshad, Adopt: An adaptive optimization framework for large-scale power distribution systems, IEEE International Conference on Self-Adaptive and Self-Organizing Systems (2009) 254–264.

[40] M. A. Marsan, L. Chiaraviglio, D. Ciullo, M. Meo, A simple analytical model for the energy-efficient activation of access points in dense wlans, in: Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy '10, ACM, New York, NY, USA, 2010, pp. 159–168.

[41] C. Cavdar, A. Yayimli, L. Wosinska, How to cut the electric bill in optical wdm networks with time-zones and time-of-use prices, in: Optical Communication (ECOC), 2011 37th European Conference and Exhibition on, 2011, pp. 1 –3.

[42] C. Peng, S.-B. Lee, S. Lu, H. Luo, H. Li, Traffic-driven power saving in operational 3G cellular networks, in: 17th MobiCom, ACM, New York, NY, USA, 2011, pp. 121–132.

[43] S. Lee, S. Sahu, Efficient server consolidation considering intra-cluster traffic, in: GLOBECOM, 2011, pp. 1–6.

[44] E. Pinheiro, R. Bianchini, E. V. Carrera, T. Heath, Load balancing and unbalancing for power and performance in cluster-based systems, in: Compilers and Operating Systems for Low Power, 2001.

[45] Y. Yao, L. Huang, A. Sharma, L. Golubchik, M. Neely, Data centers power reduction: A two time scale approach for delay tolerant workloads, in: INFOCOM, 2012 Proceedings IEEE, 2012, pp. 1431 –1439.

[46] H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, F. B. Cetin, S. Babu, Starfish: A self-tuning system for big data analytics, in: 5th Biennial Conference on Innovative Data Systems Research (CIDR) 2011, 2011, pp. 261 –272.

[47] H. Herodotou, F. Dong, S. Babu, No one (cluster) size fits all: automatic cluster sizing for data-intensive analytics, in: 2nd ACM SOCC, ACM, New York, NY, USA, 2011, pp. 18:1–18:14.

[48] D. Aikema, R. Simmonds, Electrical cost savings and clean energy usage potential for hpc workloads, in: Sustainable Systems and Technology (ISSST), 2011 IEEE International Symposium on, 2011, pp. 1 –6.

[49] S. Pelley, D. Meisner, T. F. Wenisch, J. W. VanGilder, Understanding and abstracting total data center power, Proc. 2009 Workshop on Energy Efficient Design (WEED '09).

[50] N. Padhy, Unit commitment-a bibliographical survey, Power Systems, IEEE Transactions on 19 (2) (2004) 1196–1205.

[51] A. Nazir, S. Raza, C.-N. Chuah, Unveiling facebook: a measurement study of social network based applications, in: IMC, ACM, New York, NY, USA, 2008, pp. 43–56.

[52] A. Gandhi, M. Harchol-Balter, R. Das, C. Lefurgy, Optimal power allocation in server farms, SIGMETRICS Perform. Eval. Rev. 37 (1) (2009) 157–168. doi:10.1145/2492101.1555368.
URL http://doi.acm.org/10.1145/2492101.1555368

[53] W. H. Kwon, S. H. Han, Receding Horizon Control, Elsevier, 2005.

[54] G. Box, G. M. J. , G. C. Reinsel, Time Series Analysis: Forecasting and Control, Prentice-Hall, 1994.

[55] E. Schurman, J. Brutlag, The User and Business Impact of Server Delays, Additional Bytes, and HTTP Chunking in Web Search, online, last accessed 3-April-2013 (June 2009).
URL `http://oreil.ly/fTmYwz`